

Ref. 2

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-049493

(43) Date of publication of application : 15.02.2002

(51)Int.Cl. G06F 9/445  
G06F 9/44

(21)Application number : 2000-242752

(71)Applicant : HITACHI LTD

(22) Date of filing : 04.08.2000

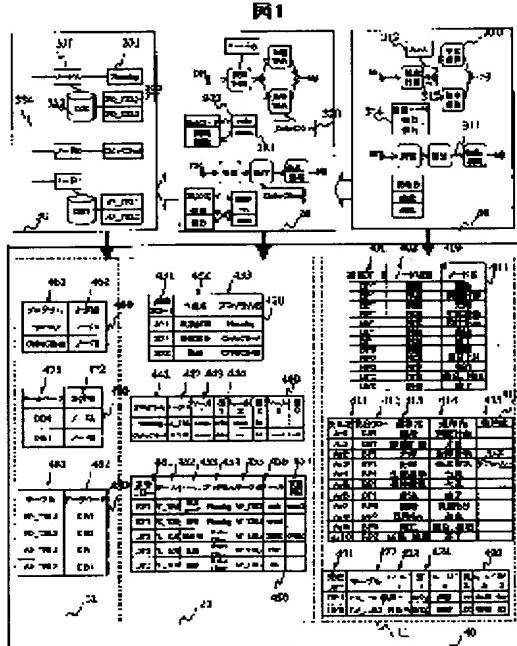
(72)Inventor : TSUKIOKA MIYOKO  
HASHIMOTO TETSUYA  
KOBAYASHI ATSUSHI  
AOKI MAKOTO  
SATO YASUSHI

(54) METHOD AND SYSTEM FOR GENERATING PROGRAM CONTROL INFORMATION, AND RECORDING MEDIUM STORED WITH PROCESSING PROGRAM THEREFOR

**(57)Abstract:**

**PROBLEM TO BE SOLVED:** To provide a method for generating flexible program control information easy in reutilization.

**SOLUTION:** An operation flow control information generation step 10, a program cooperation information generation step 20 and a system configuration information generation step 30 are separated from each other, thus the reutilization and the change of configuration are facilitated. Also definition information 11, 21, 31 acquired from respective steps 10, 20, 30 are stored in a repository 40, then the information 11, 21, 31 are integrally managed and information is obtained from relation among the definition information 11, 21, 31, thus an operating method to be easily used is obtained.



(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-49493

(P2002-49493A)

(43) 公開日 平成14年2月15日 (2002.2.15)

(51) Int.Cl.<sup>7</sup>

G 0 6 F 9/445  
9/44

識別記号

F I

G 0 6 F 9/06

テーマコード(参考)

6 1 0 A 5 B 0 7 6  
6 2 0 A

審査請求 未請求 請求項の数13 O.L (全13頁)

(21) 出願番号

特願2000-242752(P2000-242752)

(22) 出願日

平成12年8月4日 (2000.8.4)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 月岡 美代子

神奈川県川崎市幸区鹿島田890番地 株式会社日立製作所ビジネスソリューション開発本部内

(72) 発明者 橋本 哲也

神奈川県川崎市幸区鹿島田890番地 株式会社日立製作所ビジネスソリューション開発本部内

(74) 代理人 100075096

弁理士 作田 康夫

最終頁に続く

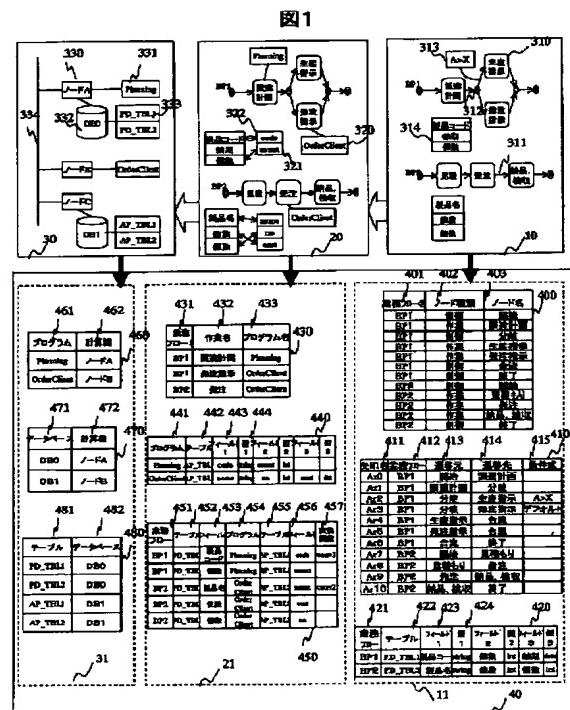
(54) 【発明の名称】 プログラム制御情報生成方法およびシステム並びにその処理プログラムを格納した記録媒体

(57) 【要約】

【課題】 本発明は、再利用しやすく柔軟なプログラム制御情報生成方法を提供することを目的とする。

【解決手段】 業務フロー制御情報生成ステップ10とプログラム連携情報生成ステップ20およびシステム構成情報生成ステップ30を分離することにより再利用や構成変更を容易とし、さらにそれぞれから取得される定義情報11, 21, 31をリポジトリ40に格納して統合的に管理し、定義情報間の関連から情報を得て、使いやすい運用方法を提供する。

【効果】 本発明により、再利用しやすく柔軟なプログラム制御情報生成方法が提供できる。



**【特許請求の範囲】**

【請求項1】(a)複数の作業とそれらの処理順序および処理順序を制御する条件を含む業務フローの定義情報と該業務フローで参照および更新される業務データの定義情報を含む業務フロー制御情報生成ステップと、

(b)業務フローにおける作業とプログラムの対応の定義情報とプログラムの入出力メッセージの定義情報と該入出力メッセージと業務データの対応の定義情報を含むプログラム連携情報生成ステップと、

(c)プログラムを該計算機システムのどの計算機に搭載するかを示す定義情報とデータベースをどの計算機に搭載するかを示す定義情報と業務データを保持するテーブルとプログラムの入出力メッセージを保持するテーブルをどのデータベースに作成するかの定義情報を含むシステム構成情報生成ステップとを有することを特徴とするプログラム制御情報生成方法。

【請求項2】請求項1の方法であり、(a)(b)(c)の3ステップを、それぞれ異なる画面で行うユーザインタフェースを有するプログラム制御情報生成方法。

【請求項3】請求項1の方法であり、(a)(b)(c)で入力された定義情報を、それぞれ複数個のテーブルに格納し、かつ(a)(b)(c)間のそれぞれの関連を示す情報もそれぞれ複数個のテーブルに格納することを特徴とするプログラム制御情報生成方法。

【請求項4】請求項1の(a)で入力された定義情報のみを取り出して業務フローテンプレートとして請求項2における画面上に表示し、それらの業務フローテンプレートの組み合わせ操作や拡張操作を含むユーザインタフェースを備え、そのユーザインタフェースによる入力を用いて請求項3における業務フロー制御情報を格納した複数個のテーブルに対する操作を行い、新たな業務フローテンプレートを生成することを特徴とするプログラム制御情報生成方法。

【請求項5】請求項2における画面上に業務フローテンプレートを表示し、請求項1における(b)から開始して異なるプログラム連携情報や異なるシステム構成情報を入力するユーザインタフェースを備え、そのユーザインタフェースを通じて入力されたプログラム連携情報やシステム構成情報と先に表示した業務フローテンプレートの情報を格納した複数テーブルからプログラム制御情報を生成することを特徴とするプログラム制御情報生成方法。

【請求項6】請求項1の(a)(b)で入力された定義情報と(a)(b)の関連を示す定義情報をあわせて取り出してプログラム連携テンプレートとして請求項2における画面上に表示し、それらのプログラム連携テンプレートの組み合わせ操作や拡張操作を含むユーザインタフェースを備え、そのユーザインタフェースによる入力を用いて請求項3におけるプログラム連携情報を格納した複数個のテーブルに対する操作を行い、新たなプログラム連携テン

プレートを生成することを特徴とするプログラム制御情報生成方法。

【請求項7】請求項2における画面上にプログラム連携テンプレートを表示し、請求項1における(c)から開始して異なるシステム構成情報を入力するユーザインタフェースを備え、そのユーザインタフェースを通じて入力されたシステム構成情報と先に表示したプログラム連携テンプレートの情報を格納した複数テーブルからプログラム制御情報を生成することを特徴とするプログラム制御情報生成方法。

【請求項8】請求項3における複数テーブルを操作し、該計算機システムに含まれる各計算機で必要となる定義情報を抽出し、各計算機に配布する情報を構成するプログラム制御情報生成方法。

【請求項9】業務フローに対する開始・終了・一時停止を含む運用指示の入力を、請求項3における複数テーブルの操作により、該計算機システム中の対応するプログラムや計算機を含むシステム構成要素を取得し、入力された運用指示に対応するコマンドを該システム構成要素で実行するプログラム制御情報生成方法。

【請求項10】請求項3における複数テーブルの操作により、該計算機システム中のプログラムや計算機を含むシステム構成要素で発生した障害や運用作業による停止を含むイベントが業務フローに与える影響範囲を特定し、その影響範囲を該業務フローとともに画面に表示するプログラム制御情報生成方法。

【請求項11】請求項1の各ステップで生成した定義情報を変更する場合に、請求項3における複数テーブルの操作により、その変更が該計算機システムに与える影響範囲を特定してプログラム制御情報に反映したり、その定義変更の影響により変更が必要となる他の定義情報の入力を促す画面を表示することを含むプログラム制御情報生成方法。

【請求項12】複数の作業とそれらの処理順序および処理順序を制御する条件を含む業務フローの定義情報と該業務フローで参照および更新される業務データの定義情報を含む業務フロー制御情報生成手段と、

業務フローにおける作業とプログラムの対応の定義情報とプログラムの入出力メッセージの定義情報と該入出力メッセージと業務データの対応の定義情報を含むプログラム連携情報生成手段と、

プログラムを該計算機システムのどの計算機に搭載するかを示す定義情報とデータベースをどの計算機に搭載するかを示す定義情報と業務データを保持するテーブルとプログラムの入出力メッセージを保持するテーブルをどのデータベースに作成するかの定義情報を含むシステム構成情報生成手段とを備えたことを特徴とするプログラム制御情報生成システム。

【請求項13】複数の作業とそれらの処理順序および処理順序を制御する条件を含む業務フローの定義情報と該

業務フローで参照および更新される業務データの定義情報と業務フローにおける作業とプログラムの対応の定義情報とプログラムの入出力メッセージの定義情報と該入出力メッセージと業務データの対応の定義情報を含むプログラム連携情報生成ステップと、  
プログラムを該計算機システムのどの計算機に搭載するかを示す定義情報とデータベースをどの計算機に搭載するかを示す定義情報と業務データを保持するテーブルとプログラムの入出力メッセージを保持するテーブルをどのデータベースに作成するかの定義情報を含むシステム構成情報生成ステップとを有するプログラム制御情報生成プログラムを格納したことを特徴とする記録媒体。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】本発明は、プログラム制御情報生成技術に係わり、特に定義された業務フローに従って関連付けたプログラムを計算機に搭載して稼動する技術に関する。

##### 【0002】

【従来の技術】既存プログラムを連携して計算機システム開発を行うEAI（エンタープライズ・アプリケーション・インテグレーション）技術が注目されている。EAI技術は、システム全体を新規に作る手法に比べ、レガシーアプリケーションや流通パッケージをそのまま利用出来、開発期間が短縮されコストが低減できることや、他システムとの連携が容易なことで、事業の多角化に伴う新サービス提供や、組織変更・企業合併により発生する計算機システムの再編成が迅速に行えるという利点を備えている。これまでのEAI技術には、ビジネスプロセス定義を用いてメンテナンス性を向上させる技術が特開平11-316780に開示されている。

##### 【0003】

【発明が解決しようとする課題】前記従来技術では、新たにプログラムを開発せずに、既存の各種プログラムを接続して動作させることを主な目的としている。そのため、メッセージの形式変換機能、プロトコル変換機能、プログラム呼び出し機能などをあらかじめ提供したり、簡易に開発できるツールを提供したりしている。しかし、システム構成全体を考えれば、全体の論理的な流れの設計やプログラムをどの計算機に搭載するか等の物理構成の設計については、新規開発と同様に毎回すべて作り直しになってしまい、再利用性に乏しいという課題がある。

##### 【0004】

【課題を解決するための手段】上記課題を解決するために、本発明では、論理的な流れの設計にあたる業務フロー制御情報生成ステップと物理構成の設計にあたるシステム構成情報生成ステップおよびその二者間を対応させるプログラム連携情報生成ステップの3段階で計算機シ

ステムを生成する方法を提供する。それぞれのステップで入力した定義情報をステップごとにいくつかのテーブルに保存することで、各ステップの定義情報をテンプレートとして抽出し、テーブル操作により組み合わせや拡張を行って別の計算機システム設計にも利用できるようになることで、上記の課題を解決する。

##### 【0005】

【発明の実施の形態】以下本発明の実施形態について図面を用いて説明する。

【0006】本発明実施形態の全体構成例を図2を用いて説明する。図中に太い線の四角形で表す11, 21, 31と、101, 111, 122が、本発明で導入するモジュールであり、他の細い線の四角形は本発明実施のために必要な既存のモジュールである。105は運用管理ノードであり、運用画面を表示するための表示装置61と運用作業の指示を入力するための入力装置63を備え、リポジトリ40に格納される定義情報（業務フロー情報11, プログラム連携情報21, システム構成情報31）をリポジトリ管理プログラム102で管理し、その情報を利用して統合運用プログラム111で統合的なシステム運用管理を行う。100は開発ノードであり、リポジトリの定義情報にアクセスするリポジトリアクセスプログラム112と設計画面を提供し定義情報をリポジトリに格納する開発環境プログラム101を搭載し、設計画面を表示する表示装置60と定義情報を入力するための入力装置60を備える。120はワーク管理ノードであり、状態管理テーブル123を保持するデータベースを備え、このテーブルを用いて業務フローの状態管理を行うワーク管理プログラム121と、運用管理ノード105から要求を受け付けたり運用管理ノードにイベントの通知を行うなど連携して運用を行う運用プログラム122を搭載する。130は業務実行ノードであり、業務を実装するアプリケーションプログラム133とメッセージの意味変換や形式変換を行い該プログラムを呼び出すアプリケーション連携プログラム132とワーク管理サーバと連携して動作するためのワーク管理クライアントプログラム131と、運用プログラム122を搭載する。140, 150も同様のプログラムを搭載する業務実行ノードであるが、業務内容に応じてアプリケーションプログラムはそれぞれ異なり、そのアプリケーションプログラムに応じてアプリケーション連携プログラムも異なる。また、140のように業務内容によってはデータベース141を備えるノードもある。上記の各ノード（計算機）はローカルエリアネットワーク70を介して接続し互いに通信する。なお、本実施例では運用管理、開発、ワーク管理、業務実行の各機能を別々のノードに配置したが、いくつかの機能を1つのノードに配置することも可能である。例えば運用管理機能と開発機能を1つのノードに配置したり、運用管理機能とワーク管理機能を1つのノードに配置したりすることが考えられ

る。また、開発機能を複数のノードに配置することも可能である。本発明で示す3段階のステップのそれぞれを各設計の専門家が担当する場合、あるいは1つのステップを複数の設計者で共同開発する場合など、開発機能を各設計者の使用している計算機全てに配置すると便利である。このような場合、リポジトリ管理プログラム102が排他制御を行って入力される定義情報の一貫性を維持する。

【0007】本発明によるプログラム制御情報生成方法について説明する。図1は業務フロー制御情報生成ステップの入力画面10とその定義情報を格納するテーブル400, 410, 420, プログラム連携情報生成ステップの入力画面20とその定義情報を格納するテーブル430, 440, 450, システム構成情報生成ステップの入力画面30とその定義情報を格納するテーブル460, 470, 480である。開発環境プログラム101が、作業、遷移矢印、分岐ノード、業務データ、プログラム、計算機、データベースなどを示すアイコンを表示し、設計者はそれらを画面上に配置し、必要に応じて名前などの文字列を入力して、10, 20, 30のような画面とする。開発環境プログラムは、これらの入力を400～480のようなテーブルとしてリポジトリ40に格納する。

【0008】10は業務フロー制御情報生成画面においてB P 1とB P 2の2つの業務フローを定義した例を示している。設計者は、開発環境プログラムが提供する業務フロー制御情報生成画面に、作業ノード310、遷移矢印311、制御ノード312、業務データ314を示すアイコンを配置し、名称や属性を入力することで、10のような画面を構成する。開発環境プログラムは入力された定義情報を解析しリポジトリに40に格納する。開発環境プログラムでは定義情報を解析する際に、入力ミスや設計ミスがないか調べ、その結果を表示し設計者に訂正を促すことも可能である。B P 1は生産指示または発注指示をする業務フローを示し、B P 2は発注の業務フローを示している。業務フローを構成する作業ノードは310のように示されその順序は遷移矢印311で接続することで示される。312に示す小さな丸は制御ノードを示しており、分岐する場合は313のように条件を指定する。400はノード名定義テーブルで、業務フローノード名401とノード種別402とノード名403のフィールドからなる。410は遷移矢印定義テーブルで各矢印の識別子411と属する業務フローノード名412、遷移元413と遷移先414および条件式415のフィールドからなる。420は業務データテーブル定義テーブルであり、業務フローノード名421、業務データを格納するテーブル名422とそのテーブルのレコードを規定するフィールド名423と型424をフィールドとして持つ。テーブル名は開発環境プログラムにより自動生成することも可能であるし、業務データの属性として設計者

が指定することも可能である。また、420では業務データを格納するテーブル(PD\_TBL1, PD\_TBL2)のフィールド数は3が最大となってしまうが、この最大値は開発環境プログラムの定義で設定が可能である。

【0009】20はプログラム連携情報生成画面においてB P 1とB P 2の2つの業務フローにプログラムを対応づけた例を示している。業務フロー制御情報生成画面10からプログラム連携情報生成20への遷移は、10の終了後ひきつづき20へと移行することも可能であるし、10終了後別の機会に業務フロー定義情報をリポジトリから取得し画面に表示させてから新たに20を開始することも可能である。プログラム連携情報生成画面20でも、10の場合と同様に、設計者はプログラムやデータ構造を示すアイコンを配置し、名称や属性を入力する。開発環境プログラムは画面10の場合と同様に、定義情報を解析してリポジトリ40に格納する。20はB P 1の調達計画と発注指示の作業、およびB P 2の発注作業にプログラムを割り当てた例である。20にはB P 1の発注指示とB P 2の発注に同じOrderCodeinent320を使うことが示されている。321はプログラムに入出力するメッセージを定義し、322はそのメッセージと10で定義した業務データとの対応を定義している。業務データとの間で意味や型の変換が必要な場合には、322の属性として変換メソッドを定義することが可能である。430は対応プログラムテーブルであり、業務フローノード名431と作業名432とプログラム連携情報生成画面で入力された対応するプログラム名433をフィールドとして持つ。440はプログラム入出力テーブル定義テーブルであり、プログラム名441とその入出力データを格納するテーブル名442とそのテーブルのレコードを規定するフィールド名443と型444をフィールドを持つ。442のテーブル名は開発環境プログラムにより自動生成することも可能であるし、プログラム入出力データの属性として設計者が指定することも可能である。また、440では入出力データを格納するテーブル(AP\_TBL1, AP\_TBL2)のフィールド数は3が最大となってしまうが、この最大値も開発環境プログラムの定義で設定が可能である。450は業務データとプログラム入出力メッセージの対応を保持するマッピングテーブルであり、451～453に業務データのフィールドを、454～456にそれに対応するプログラム入出力メッセージのフィールドを、457に322の属性として入力される、二者間の変換メソッドを、それぞれ保持する。

【0010】以上述べた第2ステップまでに定義された情報に、システム構成情報生成画面30で入力される第3ステップのシステム構成情報を加えると実際に分散計算機上で稼動させるために必要な情報が得られる。設計者は開発環境プログラムが提供する画面上に、計算機3

30, プログラム331, データベース332, テーブル333, ネットワーク接続334を示すアイコンを配置し, 名称や属性を入力することで, 30のような画面を構成する。テーブル名とプログラム名の入力の際に, プログラム連携情報生成画面20で設定されたテーブル名, プログラム名を開発環境プログラムが入力画面に提示して, 設計者が選択して指定することも可能である。460はプログラム配置テーブルでありプログラム名461とそのプログラムを搭載する計算機名462の組をレコードとして保持する。470はデータベース配置テーブルであり, データベース名471とそのデータベースが配置される計算機名472の組をレコードとして保持する。480はテーブル配置テーブルであり, テーブル名481とそのテーブルを保持するデータベース名482の組をレコードとして保持する。

【0011】次にテンプレートについて説明する。業務フロー制御情報生成画面において, 既に定義された業務フローをテンプレートとして用いて, それを拡張したり組み入れたりすることで新たな業務フローを簡単に生成することが可能である。ここで, 業務フローテンプレートとは, ノード名定義テーブル400と遷移矢印定義テーブル410と業務データテーブル定義テーブル420から, ある業務フロー名を持つレコードを取り出したものである。図3はBP1を基にして新たな業務フローBP3を生成する画面の例である。BP3は, BP1の発注指示のあとに, BP2を挿入したものである。図4は, 設計画面での入力とその際に定義テーブルを変更する手続きを示している。これにしたがって図3の画面で示される実施例を説明する。

【0012】まず, 図3の画面においてBP1を使って新しい業務フローBP3を定義することを指定する600。画面にはBP1と同じ図がBP3という名前になって表示される。ノード名定義テーブル400, 遷移矢印定義テーブル410には, それぞれBP1のレコードをコピーしBP名をBP3と変更した新規レコードを追加する601。画面では次に, BP3の発注指示354の後, 合流355の前に, BP2をあらわすアイコン357を挿入し, BP2をBP3のこの位置に組み入れることを指定する602。このとき, 図5に示すように, ノード名定義テーブル400には, BP名=BP2であるレコードのうち, 開始ノードと終了ノードをあらわすレコード以外のレコードをコピーし, BP名=BP3として追加する603。図6に示すように, 遷移矢印定義テーブル410にも同様にBP名=BP2のレコードをコピーしBP名=BP3として追加し, さらにBP2の挿入を反映して矢印の遷移先を2個所変更する605。この実施例においては, 「発注指示」と「合流」の間にBP2を挿入したので, 遷移元ノード「発注指示」のレコードの遷移先ノード名を, BP2の開始ノード直後のノード「見積もり」に変更し, BP2の終了ノード直前のノード「納品・検収」を, 「合流」に変更する。

ノード「納品・検収」を, 「合流」に変更する。

【0013】次に, 設計画面でBP3の業務データを入力する606。BP1をテンプレートとして使用することを選択した時, BP1と同じ業務データが画面に表示される。BP2を組み入れることを入力した時に, BP2の業務データもマージして画面に表示し, 業務データテーブル定義テーブル420にも, BP1とBP2のレコードをマージしたレコードをBP名=BP3として挿入する。本実施例では, 製品コードと製品名が同じものをあらわすと設計者が判断し, 製品コードのみ残して361で確定し入力した状況606を示している。このように変更を加える場合には, 業務データテーブル定義テーブル420にもそれを反映する。この場合は, BP3としてBP1とBP2のフィールド名とその型のリストをマージしたものから, 「フィールド名=製品名」とその型を除いたリストを持つレコードが追加されることになる607。以上のようにして, 業務フローテンプレートから, それを用いて新たに定義された業務フローに対応するレコードが生成される。

【0014】プログラム連携テンプレートについても, 上述の業務フローテンプレートと同様に, 既に定義済みのプログラム連携情報をテンプレートとして利用することが可能である。プログラム連携テンプレートは, 業務フローテンプレートの内容に加えて, 対応プログラムテーブル430とプログラム入出力テーブル定義テーブル40とマッピングテーブル450のレコードを持つ。業務フロー定義の部分に関しては, すでに説明した図4の手続きで定義テーブルにレコードを追加する。対応プログラム関連付けテーブル430とプログラム入出力テーブル定義テーブル440は, 単にレコード追加だけを行う。マッピングテーブル450は, 業務フローテンプレートにおける業務データテーブル定義テーブル420と同様に, 使用するプログラム連携テンプレートの定義をマージしたものが表示され, そのまま確定されればレコードもマージしたものをそのまま追加する。設計者が業務データの変更などによりマッピングを変更した場合は, それを反映してレコードも変更したものを追加する。

【0015】以下に運用管理機能について実施例を用いて説明する。

【0016】まず, 運用管理機能のうち実行環境を構築するための定義情報の配布について説明する。図7は業務実行ノードに業務データテーブル定義を配布し各業務実行ノードでテーブルを生成する手続きを示している。220では業務データテーブル定義テーブル420よりBP1のレコードを選択しテーブル名(PD\_TBL1)とフィールド名とその型からなるスキーマ情報(製品コード: String, 個数: int, 納期: date)を取得する。221ではテーブル配置テーブル480によりPD\_TBL1がデータベースDB0に配置さ

れることを得る。222ではデータベース配置テーブル470によりDB0がノードAに配置していることを得る。そこで223でノードAへ220で取得したテーブル名とスキーマ情報を送付し、224ではノードAの運用プログラムがこれを受け取り、該スキーマにキーとして案件IDのフィールドを加えて、テーブルPD\_TBL1を生成する。図8の510は、このようにしてノードAのDB0上に生成される業務データテーブルPD\_TBL1の例である。BP2の業務データテーブルPD\_TBL2も同様にノードAのデータベースDB0上に図9の520のように生成される。また、プログラム入出力メッセージについても、プログラム入出力テーブル定義テーブル440を用いて、業務データと同様の手続きにより生成される。図10の530と図11の540はノードCのDB1上に生成されるプログラム入出力テーブルAP\_TBL1とAP\_TBL2の例である。ここまで、データベース配置ノードにテーブル定義情報を配布してテーブルを生成する手続きの実施例を示したが、同様にして各プログラムの配置ノードに実行環境作成に用いる定義情報を配布することが可能である。その場合に配布する定義情報としては、各プログラムに設定する定義項目のうち、例えばサーバ名やポート番号などシステム内で一意にしたい項目や、システム設計者があらかじめ決めるべき項目などがあり、これらをシステム構成情報生成ステップにおいてシステム設計者が入力し、それを配布することが可能である。他に、業務フロー制御情報生成ステップやプログラム連携情報生成ステップで決定される項目でも、上記説明と同様に各ステップで入力された情報間の関連を用いて、実行環境で必要な情報をテーブル操作により取得することで、業務実行ノードに配布することが可能となる。

【0017】次に業務フローを対象として運用情報の表示や運用作業の指示を行う、業務観点での運用管理機能について説明する。図13の実行状態テーブル500は、実行環境の状態表示を各情報生成ステップで定義された情報と結び付けて行うために設けるものであり、リポジトリ40に配置され統合運用プログラム111がこれを管理する。ここに示す例では状態としてプログラムの実行状態を示すフィールドのみを保持しているが、他に計算機の稼動状態やネットワークの通信状態などを示すフィールドを含むことも可能である。まず、実行状態テーブルに格納するデータの生成手続きを、業務フローBP1を例として説明する。図12は業務フローBP1に関係するレコードを生成し実行状態テーブルに挿入する手続きを示している。210ではBP1に関わるプログラム名の取得を、対応プログラムテーブル430において、業務フロー名=BP1であるレコードを選択しそのプログラム名を取得することで行う。211ではプログラム配置テーブル460でそのプログラムが搭載されている計算機名を取得する。業務フローからみれば、こ

こで取得された計算機はBP1を実行する計算機である。212では210と211により取得したプログラム名と計算機名を1レコードとして実行状態テーブル500に挿入する。ここで、実行状態フィールド504は初期値(NULLなど)としておく。業務実行ノードから稼動状況が報告されると、統合運用プログラム111が実行状態フィールドの値を更新する。稼動状況の報告は一定時間毎に行うことも可能であるし、状態が変化したときのみ行うことも可能である。業務フロー制御情報生成画面10で設計者が作成した業務フローの図を、運用管理画面に表示し、作業を示すアイコンを選択して状態表示ボタンを押すと、実行状態テーブルを参照して該作業を実行するプログラムの状態を取得し、それを画面上に表示することが可能である。また、業務フローを選択して状態表示ボタンを押すことにより、該業務フローに関連するすべてのプログラムの状態を表示させることも可能である。

【0018】さらに、状態変化がプログラムや計算機の障害であった場合、定義情報を利用してその影響範囲を求め、運用管理画面に直ちに表示することも可能である。以下その方法について、計算機障害が発生した場合を例として図14のフローを用いて説明する。230では、実行環境の運用機能が障害を検知し、これを運用管理ノードの統合運用プログラム111に通知する。231で障害を通知された統合運用プログラムはまずプログラム配置テーブル460を参照して、障害が発生した計算機で実行されていたプログラム名を取得する。232では更に対応プログラムテーブル430を参照して該プログラムと対応する作業名と業務フロー名を取得する。以上の手続きにより、障害の発生した計算機で処理されていた業務フロー名と作業名が求められた。233でこれを運用管理画面に表示する。図15に示すように、業務フロー制御情報生成画面10で作成された図上に表示することも可能である。図15では、発注作業を実現するプログラムOrderClientプロセスが発注処理307において障害を発生した場合に、BP2のみが影響をうけるのではなく、同じOrderClientが実現している発注指示作業304も影響を受けることを示している。また、上記の例では、計算機障害とプログラム障害に着目したが、データベースについても同様に障害の影響範囲を求めることが可能である。すなわち、ノードA331に障害が発生した場合ならば、ノードAにはデータベースDB0339が配置されているので、そのDB0が保持している業務データテーブルPD\_TBL1とPD\_TBL2へのアクセスが不可となる。よって、BP1とBP2の双方が影響を受けることが導かれる。また、障害だけでなく、定義を変更したときの影響範囲も同様にテーブルを辿って求めることが出来る。例えば、業務フロー制御情報生成画面10においてBP1から「発注指示」が削除された場合、ノード名定

義テーブル400と遷移矢印定義テーブル410の対応するレコードが削除または書き換えられ、さらにプログラム連携情報生成画面20の「発注指示」およびそれと結び付けられた「Order Client」の削除と対応プログラムテーブル430の対応するレコードの削除が実行される。あるいは変更が必要な情報を更新させるような表示をすることも可能である。例えば、システム構成情報生成画面30においてOrder Clientが削除される場合、プログラム連携情報生成画面20のOrder Clientも削除されなくてはならないので、その代わりのプログラム名を入力するように促す画面を表示して設計者に新しく割り当てるプログラム名を入力させるなどである。

【0019】以下実施例における本発明の効果を述べる。

【0020】第一の効果は、開発の効率化がはかれるという点である。計算機システムの設計を3段階の情報生成ステップに分けたことから、それぞれの設計担当者を別々にすることが可能となり、設計者は自分の専門領域の設計のみに従事することが可能となるからである。

【0021】第二の効果は、設計知識の蓄積と再利用が促進されることである。業務フロー制御情報生成ステップでの定義情報のみ取り出して業務フローテンプレートとすること、業務フロー制御情報生成ステップとプログラム連携情報生成ステップの定義情報を取り出してプログラム連携テンプレートとすることにより、計算機システムを開発するたびに必然的に知識が蓄積されるからである。

【0022】第三の効果は、運用管理作業の軽減である。業務フローと実際の計算機とを設計時に入力された定義情報を用いて関連付けて管理することにより、業務観点での運用作業が可能となること、また、障害や定義変更の影響範囲を特定し提示することによる。

【0023】

【発明の効果】以上述べたように本発明では、計算機システムに搭載するプログラム制御情報の生成を3段階のステップに分けて行い、それぞれのステップで入力した定義情報をステップ毎にいくつかのテーブルに保存することで、各ステップの定義情報をテンプレートとして抽出し、テーブル操作により組み合わせや拡張を行って別の計算機システム設計にも利用できるようにすることで再利用性の高いプログラム制御情報生成方法を提供可能となった。

【図面の簡単な説明】

【図1】3段階の情報生成ステップと各ステップから入力されリポジトリに保持される定義情報を示す図。

【図2】実施形態のシステム構成図。

【図3】テンプレートを用いた業務フロー制御情報生成画面例。

【図4】テンプレートを用いた業務フロー制御情報生成

手続き。

【図5】テンプレートを用いた業務フロー制御情報生成におけるノード名定義テーブルへのレコード追加方法。

【図6】テンプレートを用いた業務フロー制御情報生成における遷移矢印定義テーブルの変更方法。

【図7】業務データテーブルのテーブル名とスキーマ情報を配布しテーブルを生成する手続き。

【図8】業務データテーブル。

【図9】業務データテーブル。

【図10】プログラム入出力テーブル。

【図11】プログラム入出力テーブル。

【図12】実行状態テーブルに1つの業務フローのレコードを追加する手続き。

【図13】実行状態テーブルの例。

【図14】障害影響範囲を特定する手続き。

【図15】運用管理画面に障害影響範囲を表示した例。

【符号の説明】

10, 20, 30 : 情報生成画面

11, 21, 31 : 各情報生成画面から入力される定義情報

40 : リポジトリ, 60, 61 : 表示装置, 62, 63 : 入力装置, 70 : ローカルエリアネットワーク, 100 : 開発ノード, 101 : 開発環境プログラム, 102 : リポジトリ管理プログラム, 105 : 運用管理ノード, 111 : 統合運用プログラム, 120 : ワーク管理ノード, 121 : ワーク管理プログラム, 122 : 運用プログラム, 123 : 状態管理テーブル, 130 : 業務実行ノード

131 : ワーク管理クライアントプログラム, 132 : アプリケーション連携プログラム, 133 : アプリケーションプログラム, 140 : 業務実行ノード, 141 : 業務データベース, 150 : 業務実行ノード 210 ~ 212 : 実行状態テーブルにレコードを追加する手続き

220 ~ 224 : 業務データテーブル生成手続き

230 ~ 233 : 障害影響範囲の表示手続き

302 ~ 308 : 障害影響範囲を表示する運用管理画面の例

310 ~ 334 : 各情報生成画面の要素

310 : 作業, 311 : 遷移矢印, 312 : 制御ノード, 313 : 分岐条件, 314 : 業務データ, 320 : プログラム, 321 : プログラム入出力メッセージ, 322 : マッピング, 330 : 計算機, 331 : プログラム, 332 : データベース, 333 : 業務データテーブル, 334 : ローカルエリアネットワーク 350 ~ 370 : 業務フローテンプレートを用いた業務フロー制御情報生成画面 400 ~ : リポジトリに格納する定義情報 400 : ノード名定義テーブル, 410 : 遷移矢印定義テーブル, 420 : 業務データテーブル定義テーブル, 430 : 対応プログラムテーブル, 440 : プログラム入出力テーブル定義テーブル, 450 : マッピングテーブル

ブル, 460:プログラム配置テーブル, 470:データベース配置テーブル, 480:テーブル配置テーブル  
500~:システム構築時に定義情報から生成されるテーブル

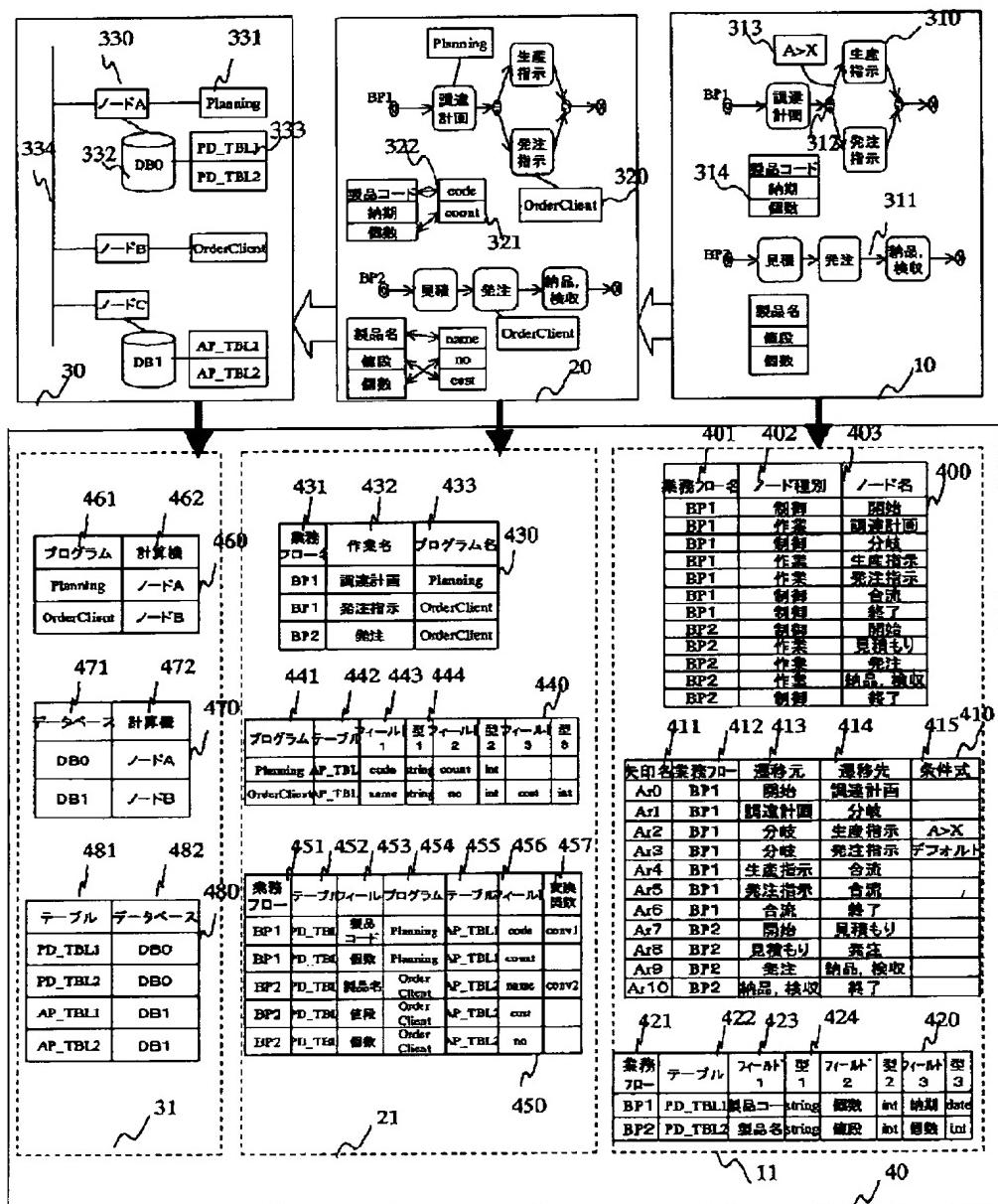
500：実行状態テーブル、510、520：業務デー

ターブル, 530, 540: プログラム入出力テーブル

## 6.0.0～：ビジネスプロセステンプレートを用いた設計時の処理

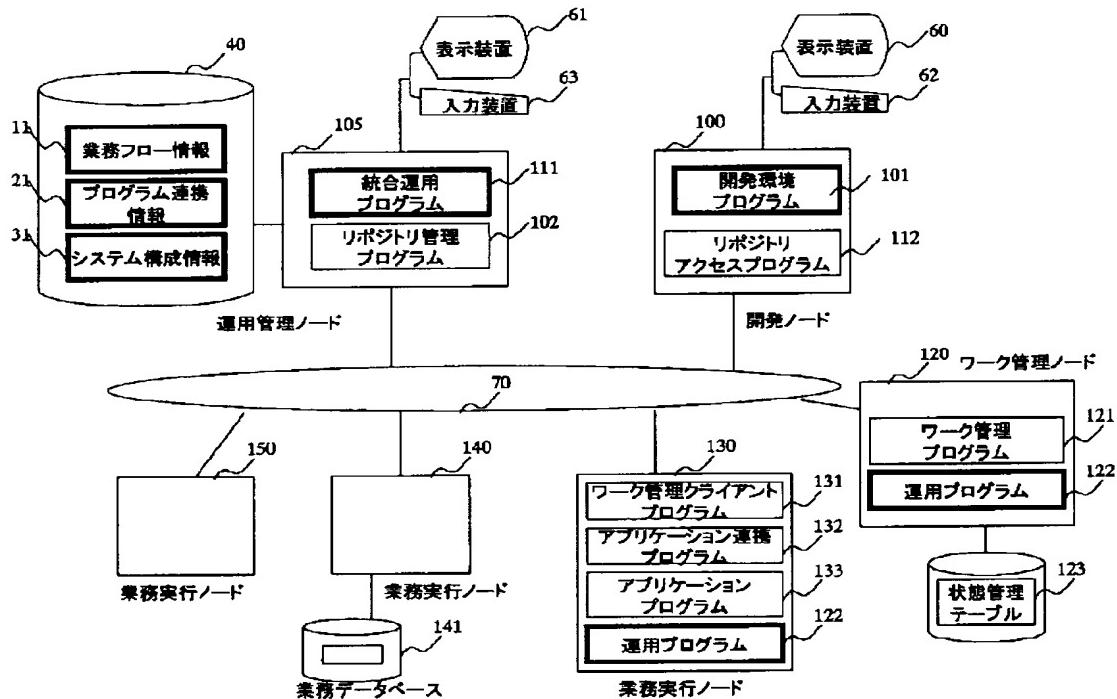
【図1】

1



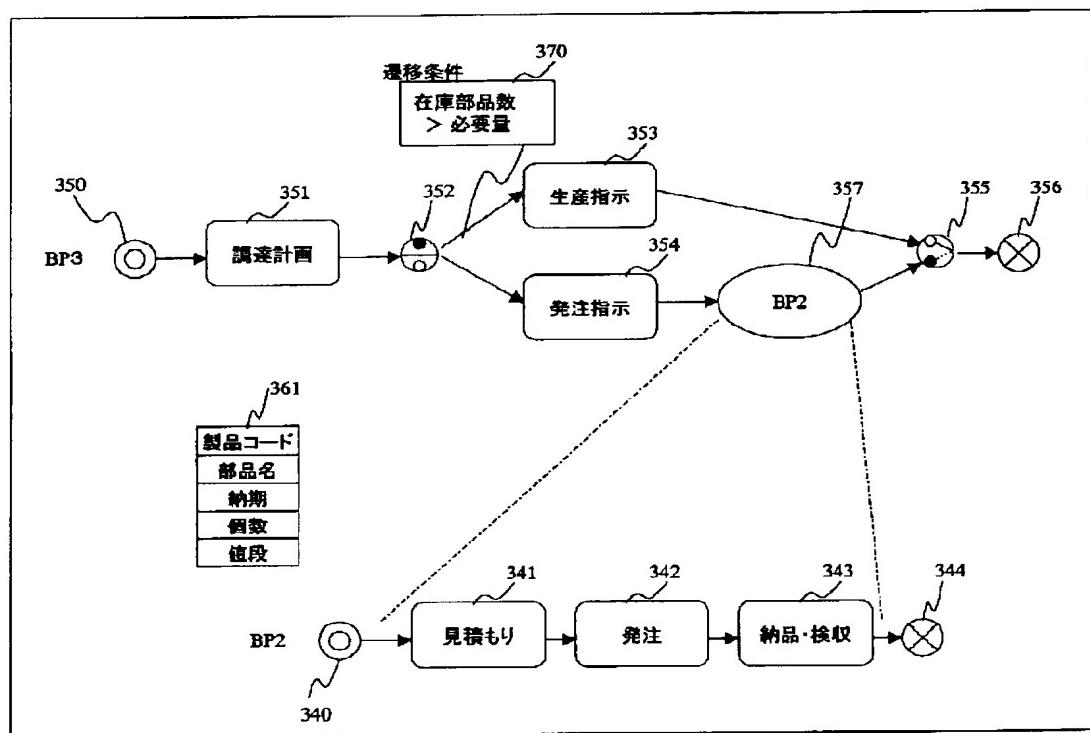
【図2】

図2



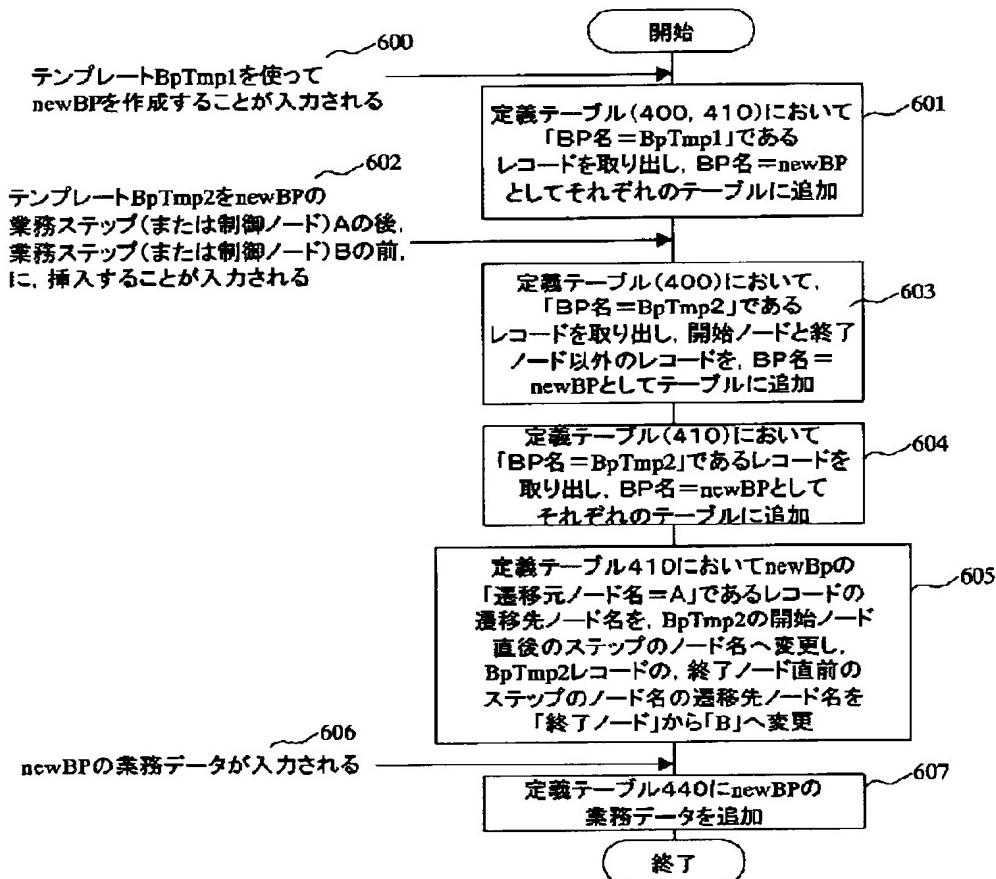
【図3】

図3



【図4】

図4



【図8】

PD_TBL1			
案件ID	製品コード	個数	納期
ID3	P1002	200	2000/02/03
ID5	P2401	50	2000/02/05

図8

【図9】

PD_TBL2			
案件ID	製品名	値段	個数
ID0	Pro11	12,700	150
ID2	Pro35	28,900	120

図9

【図10】

AP_TBL1		
案件ID	code	count
ID0	P105	20
ID1	P002	120

図10

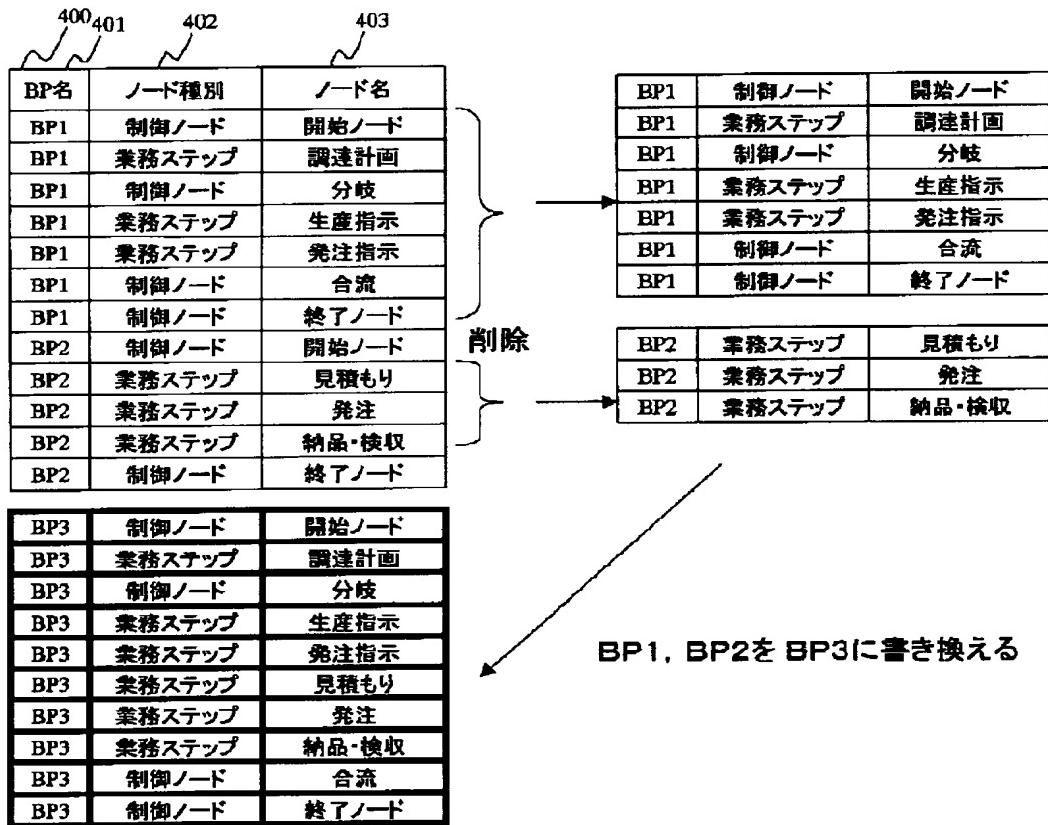
【図11】

AP_TBL2			
案件ID	name	no	cost
ID4	P004	130	12200
ID5	P017	190	2500

図11

【図5】

図5



【図13】

図13

Table showing the status of processes across different nodes (ノードA, ノードB).

**Table Headers:**

- 計算機名 (Computer Name)
- プログラム名 (Program Name)
- 業務フローネ名 (Business Flow Name)
- 実行状態 (Execution Status)

**Data Rows:**

- ノードB, OrderClient, BP2, 停止中
- ノードA, Planning, BP1, 実行中
- ノードB, OrderClient, BP1, 停止中

【図6】

図6

The diagram illustrates the transformation of a business process table from two parallel paths (BP1, BP2) to a single path (BP3).

**Original Table (図6):**

矢印名	業務	遷移元ノード名	遷移先ノード名	条件式
Ar1	BP1	開始ノード	調達計画	
Ar2	BP1	調達計画	分岐	
Ar3	BP1	分岐	生産指示	在庫部品数>必要数
Ar4	BP1	分岐	発注指示	デフォルト
Ar5	BP1	生産指示	合流	
Ar6	BP1	発注指示	合流	
Ar7	BP1	合流	終了ノード	
Ar8	BP2	開始ノード	見積もり	
Ar9	BP2	見積もり	発注	
Ar10	BP2	発注	納品・検収	
Ar11	BP2	納品・検収	終了ノード	

**Transformed Table (図6):**

矢印名	業務	遷移元ノード名	遷移先ノード名	条件式
Ar1	BP1	開始ノード	調達計画	
Ar2	BP1	調達計画	分岐	
Ar3	BP1	分岐	生産指示	在庫部品数>必要数
Ar4	BP1	分岐	発注指示	デフォルト
Ar5	BP1	生産指示	合流	
Ar6	BP1	発注指示	見積もり	
Ar7	BP1	合流	終了ノード	
Ar9	BP2	見積もり	発注	
Ar10	BP2	発注	納品・検収	
Ar11	BP2	納品・検収	合流	

**Final Table (図7):**

Ar1	BP3	開始ノード	調達計画	
Ar2	BP3	調達計画	分岐	
Ar3	BP3	分岐	生産指示	在庫部品数>必要数
Ar4	BP3	分岐	発注指示	デフォルト
Ar5	BP3	生産指示	合流	
Ar6	BP3	発注指示	見積もり	
Ar7	BP3	合流	終了ノード	
Ar9	BP3	見積もり	発注	
Ar10	BP3	発注	納品・検収	
Ar11	BP3	納品・検収	合流	

矢印名	業務	遷移元ノード名	遷移先ノード名	条件式
Ar1	BP1	開始ノード	調達計画	
Ar2	BP1	調達計画	分岐	
Ar3	BP1	分岐	生産指示	在庫部品数>必要数
Ar4	BP1	分岐	発注指示	デフォルト
Ar5	BP1	生産指示	合流	
Ar6	BP1	発注指示	見積もり	
Ar7	BP1	合流	終了ノード	
Ar9	BP2	見積もり	発注	
Ar10	BP2	発注	納品・検収	
Ar11	BP2	納品・検収	合流	

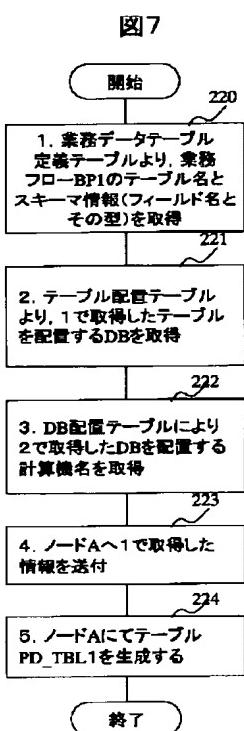
「合流」 → 「見積もり」  
 削除  
 「終了ノード」 → 「合流」

BP1, BP2をBP3に書き換える

Ar1	BP3	開始ノード	調達計画	
Ar2	BP3	調達計画	分岐	
Ar3	BP3	分岐	生産指示	在庫部品数>必要数
Ar4	BP3	分岐	発注指示	デフォルト
Ar5	BP3	生産指示	合流	
Ar6	BP3	発注指示	見積もり	
Ar7	BP3	合流	終了ノード	
Ar9	BP3	見積もり	発注	
Ar10	BP3	発注	納品・検収	
Ar11	BP3	納品・検収	合流	

【図7】

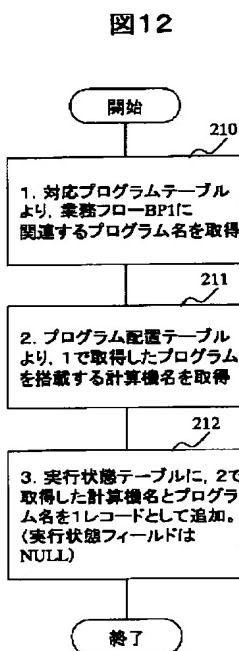
【図12】



テーブル名: PD\_TBL1  
スキーマ:  
フィールド1 : 製品コード, string

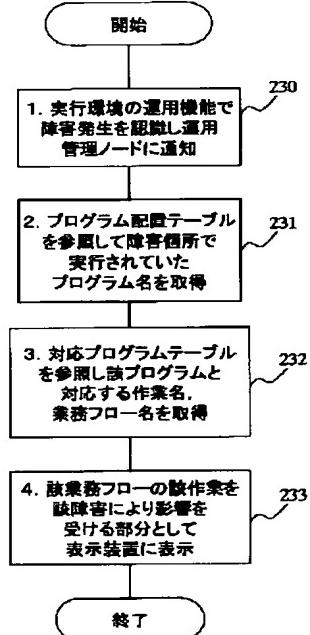
DBO取得

ノードAを取得



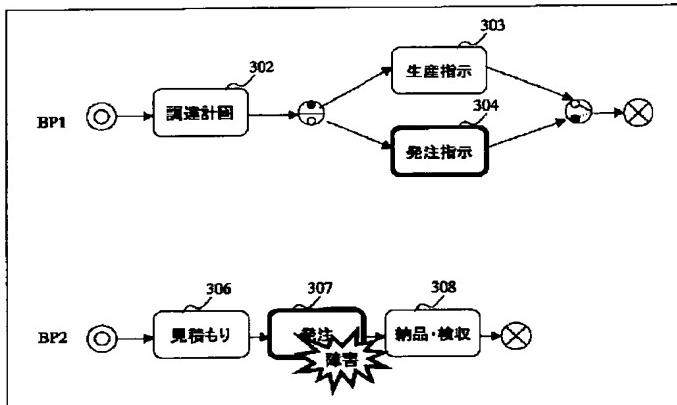
【図14】

図14



【図15】

図15



フロントページの続き

(72)発明者 小林 敦

神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア事業部内

(72)発明者 青木 誠

神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア開発本部内

(72)発明者 佐藤 康史

神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア事業部内

F ターム(参考) 5B076 AA02 AA04 AA05 AA06 AA12

DD01